DOCUMENT RESUME

ED 205 206 IR 009 510

AUTHOR Seidman, Robert H.
TITLE The Effects of Learning a Computer Programming Language on the Logical Reasoning of School Children.
PUB DATE 14 Apr 81
NOTE 63p.: Paper presented at the Annual Meeting of the American Educational Research Association (Los Angeles, CA, April 14, 1981).

EDRS PRICE MF01/PC03 Plus Postage.
DESCRIPTORS Achievement Gains; Analysis of Covariance; Analysis of Variance; Cognitive Processes; Computer Science Education; Elementary Education; Grade 5; Learning Processes; *Logical Thinking; *Microcomputers; *Programing Languages
IDENTIFIERS *LOGO System

ABSTRACT

The research reported in this paper explores the syntactical and semantic link between computer programming statements and logical principles, and addresses the effects of learning a programming language on logical reasoning ability. Fifth grade students in a public school in Syracuse, New York, were randomly selected as subjects, and then randomly placed in either the experimental or the control group. The experimental group was taught LOGO, a programming language developed for use with young children, while the control group received no special instruction. At the end of the treatment period, both groups were administered a series of tests measuring their conditional reasoning abilities. Tests were scored in two distinct ways, and the two groups were statistically compared within both scoring schemes by split-plot two-factor repeated measures and one-way analysis of variance. It was found that students in the experimental group who interpreted conditional logic statements biconditionally performed significantly better on the inversion fallacy principle than the control group; no significant difference was found when test items were scored in the traditional manner. Comparison of pre- and post-experiment achievement test scores showed a significant improvement in reading only for the control group. Some areas for further research are suggested, and a 64-item bibliography is attached. (MER)

THE EFFECTS OF LEARNING A COMPUTER PROGRAMMING LANGUAGE ON THE LOGICAL REASONING OF SCHOOL CHILDREN

Robert H. Seidman

Department of Technology & Society
State University of New York at Stony Brook

American Educational Research Association
Annual Meeting
Los Angeles, California
April 14, 1981

TABLE OF CONTENTS

I

INTRODUCTION

The central question addressed in the research reported here is: what are the effects, if any, of learning a computer programming language upon the learner's ability to reason logically?

Logical reasoning is an important cognitive attribute of central importance in many aspects of life. There is fairly widespread agreement among educators and the general public that the development of problem solving and critical thinking abilities in students are two major goals of the educational system.[1] It has been argued, quite persuasively, that logical reasoning (i.e., the drawing of valid inferences and correctly judging whether or not a statement follows from others necessarily) is central to both problem solving and critical thinking.[2] Indeed, it is difficult to imagine an approach to problem solving that does not include some aspect of deductive or inductive reasoning. Can learning a computer programming language influence the learner's ordinary-language logical reasoning ability?[3]

An empirical test of this question is vital since the rate at which microcomputers are being purchased for home and school use is increasing rapidly and is not likely to abate for at least a decade[4]. In addition, the teaching of computer programming has moved downward in the school curriculum from college to elementary school and shows signs of becoming widespread.[5]

If research should show that logical reasoning ability is enhanced by learning a computer programming language, then there would exist a compelling practical incentive for incorporationg computer programming into school curricula at all levels. In addition, this method of (indirect) instruction in logic could prove to be a viable alternative to the kind of direct logic instruction which has been largely ineffectual in the schools.[6]

However, should empirical studies show that learning a computer programming language results in learning "incorrect" logical principles, then a warning can be sounded: learning computer programming can confound attempts to teach logical reasoning and thus negatively affect the learner's problem solving and critical thinking abilities. Results such as these might lead to a collaborative effort between educators and computer scientists to redesign certain aspects of computer programming languages.

Conditional logic statements play a significant role in discourse associated with such things as explanation and argumentation. Also, laws, causal relationships, possibilities, as well as evidence relationships are frequently expressed in terms of logical statements (particularly conditional logic). In addition, the development of logical structures is a key notion in Piaget's theory of cognitive development and it is thought that the ontogenesis of logical necessity reflects the development of these structures.[7]

The use of computers in our society, both in number and scope, has grown tremendously since the advent of microcomputers in 1975. Inexpensive microcomputers ("personal computers") now cost as little as $200 and are appearing in homes and schools in increasing numbers. Between 1975 and 1979, over 500,000 personal computers were sold in the United States with an equal number produced in 1980 alone. One technology assessment predicts that 3 million units will be sold to first-time buyers in 1985 and that by 1990 a cummulative sales total of 9 million will be reached.[8]

While some computer use is undoubtedly for entertainment purposes, much of it is explicitly for educational purposes and is marketed as such.[9] The low cost of these machines is making computer programming possible for a large number of persons and the educational system is responding by purchasing a great many machines and by offering an increasing number of courses in computer programming at all levels of the system.

Some educators and policy planners suggest that proficiency in computer programming should rank with reading, writing and arithmetic literacy. One implication is that computer programming should be a requisite for grade advancement beginning as early as Kindergarten. Thus, the kind of research reported here is both timely and vital.[10]

Section II shows how certain computer programming language statements are syntactically and semantically (dis)similar to ordianry-language conditional statements. Depending upon the user's interpretation of these ordinary-language conditional statements, the computer statements constitute "correct" or "incorrect" indirect instruction in the validity and fallacy principles of conditional logic.

Section III presents an experiment that attempts to answer the central question and which demonstrates the importance of carefully assessing the logical standards against which logical test items are scored as "correct" and "incorrect." Section IV consists of conclusions drawn from this study and a discussion of the results.

I know of no previously published research that demonstrates the syntactical and semantic link between computer programming statements and logical principles. Nor do I know of research that addresses the effects of learning a computer programming language on logical reasoning ability.[11]

II

CONDITIONAL LOGIC PRINCIPLES AND COMPUTER PROGRAMMING

A. Conditional Logic Principles

By "reason logically" I mean, utilize correctly logical principles. Conditional logic is a type of propositional logic that uses the logical connectives, "if," "if-then," "only if," and "if and only if" to connect antecedent and consequent ordinary-language propositions (represented here by "p" and "q," respectively). I focus upon the "if-then" connective since "if p then q" (called the logical conditional statement) is a fairly commonly used locution in ordinary discourse. In addition, it is often used to reflect deductive necessity: a conclusion follows necessarily from a premise or set of premises by virtue of the formal structure of the premise(s) and conclusion without appeal to empirical evidence.

It is logical convention that the binary truth function representing the conditional statement renders the statement true for all truth value combinations of its antecedent and consequence except when p is true and q is false. This interpretation of the conditional statement is called material implication (or material conditional).[12] Four principles of conditional logic are of interest here. They can be viewed as corresponding to four conditional arguments.

Consider the conditional statement as the first premise in a two premise argument which has as its second premise the affirmation or denial of either the antecedent or consequence of the first premise. The conclusion of the argument is either the affirmation or denial of the proposition which is not present in the second premise. Of these four possible arguments two are valid arguments (i.e., the conclusion follows logically from the premises) and two are invalid (i.e., the conclusion does not follow logically from the premises). The arguments and the conditional principles corresponding to them are (given the conditional statement, "if p then q," as the first premise) shown in Figure 1.[13] I use "p" and "q" as shorthand notation. Everyday usage might find "it rains today" for "p" and "Mary carries her umbrella" for "q." Thus, the shorthand "if p, then q" stands for: "If it rains today, then Mary carries her umbrella."

1. Forward Conditional. The affirmation of the antecedent (p) implies the affirmation of the consequent (q).

2. Inversion. The denial of the antecedent ($\bar{p}$) does not by itself imply the denial of the consequent ($\bar{q}$).

3. Conversion. The affirmation of the consequent (q) does not by itself imply the affirmation of the antecedent (p).

4. Contraposition. The denial of the consequent ($\bar{q}$) implies the denial of the antecedent ($\bar{p}$).

Figure 1. Four Principles of Conditional Logic under the Material Conditional Interpretation

Principles 1 and 4 are the validity principles, which means that a particular conclusion follows necessarily from an argument containing the conditional statement. Principles 2 and 3 are the fallacy principles because no valid conclusion can be drawn from an argument containing the conditional statement.

Figure 1 reflects the material conditional interpretation of the logical conditional statement, "if p then q." However, empirical evidence shows that many children in the concrete operational stage of cognitive development (ages 7-11) and formal operational adolescents interpret the conditional statement in a biconditional manner (Seidman, 1980a, 1980b). Under this interpretation, the fallacy principles in Figure 1 become valid. See Figure 2.

2a. Biconditional Inversion. The denial of the if-part ($\bar{p}$) of the "if p, then q" conditional statement implies the denial of the then-part ($\bar{q}$).

3a. Biconditional Conversion. The affirmation of the then-part (q) of the "if p, then q" conditional statement implies the affirmation of the if-part (p).

Figure 2. Biconditional Interpretation of the "Fallacy" Principles of Conditional Logic

The biconditional locution is, "p if and only if q." By logical convention, the biconditional statement is true when the truth value of "p" and "q" are the same. Otherwise, the statement is false.

B. Standard Conditional Branch Statement and Logical Principles

All higher-level computer programming languages contain conditional branch statements. Ordinarily, statements (instructions) within a computer program are executed in an invariant sequence unless an instruction is encountered which commands otherwise. The conditional branch statement is such an instruction. Very simply, the conditional branch statement is the programmer's way of instructing the computer to alter the natural flow of program execution, depending upon the truth-value (truth-status) of a predicate (or propositon). This is an absolutely vital instruction without which a computer programming language would be rendered virtually useless.

The syntax of conditional branch statements in different computer programming languages differs but their logical structures are similar. The general logic of a conditional branch statement is illustrated diagrammatically in Figure 3.

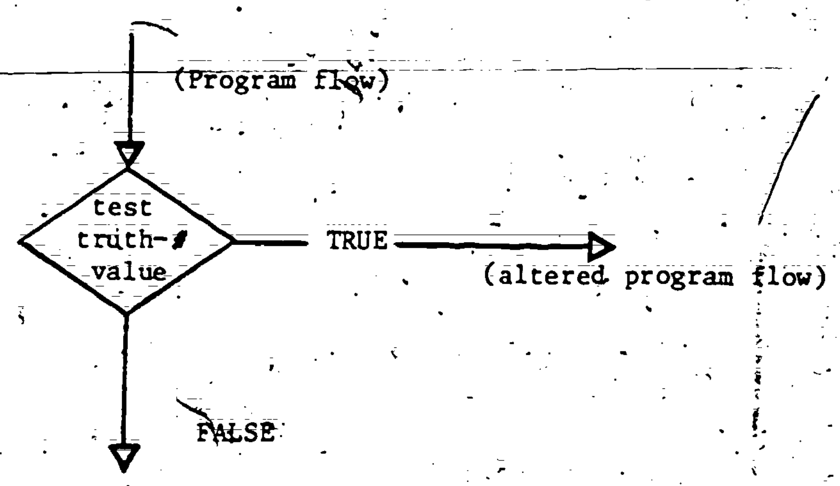

Figure 3. Logic of a Conditional Branch Statement
(tests truth-value of proposition)

The standard conditional branch statement[14] can be expressed in its general form as:

IF a THEN b (1)

where "a" is a logical expression (proposition) which evaluates to the truth value TRUE or FALSE. I call "a" the antecedent-proposition. "b" is an expression, but not a proposition. Typically, "b" causes some action to be carried out by the program such as assigning a value to a variable, branching to another program statement, output of data, or halting the execution of the program. I call "b" the consequent-action. The general logic of (1) is illustrated in Figure 4.

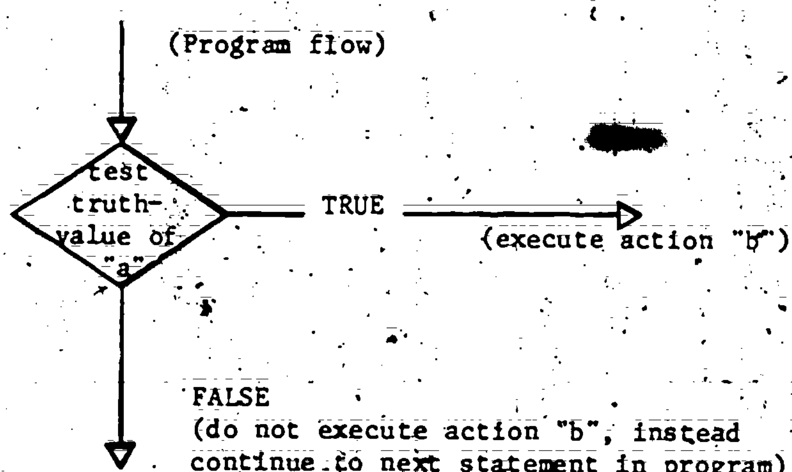

Figure 4. Logic of the Standard Conditional Branch Statement

The expression "b" is not strictly a proposition (it does not evaluate to a TRUE or FALSE truth-value). However, depending upon the truth-status of "a," it is either true or false that action "b" is carried out. In this sense, we can view "b" as quasi-propositional. It is doubtful that in practice such a sophisticated distinction is made. More likely, the programmer views "b" as true (occurring) or as false (not occurring). I use "true" and "false" in this manner, in the context of consequent-action, throughout this discussion.

There are two primary semantic properties associated with the standard conditional statement:

1. When the antecedent-proposition is true ("a"), the consequent-action will causally be true ("b"). Thus, "a" and "b" can co-occur.

2. When the antecedent-proposition is false ("$\bar{a}$"), the consequent-action will causally be false ("$\bar{b}$"). Thus, "$\bar{a}$" and "$\bar{b}$" can co-occur.

Two derivative semantic properties can be drawn from these primary properties:

3. If we know that the consequent-action is true ("b"), then we are entitled to conclude that the antecedent-proposition is not false (i.e., is true, "a"). Thus, "$\bar{a}$" and "b" cannot co-occur.

4. If we know that the consequent-action is false ("$\bar{b}$"), then we are entitled to conclude that the antecedent-proposition is false (i.e., not true, "$\bar{a}$"). Thus, "a" and "$\bar{b}$" cannot co-occur.

These standard conditional branch statement properties are illustrated in Table 1.

TABLE 1

CO-OCCURRENCES FOR "a" AND "b"

Antecedent-Proposition	Consequent-Action	Co-occurrence
a	b	T
a	$\bar{b}$	F
$\bar{a}$	b	F
$\bar{a}$	$\bar{b}$	T

Note: "T" stands for a co-occurrence and "F" stands for a non-occurrence.

Properties 1 and 4 of the standard conditional branch statement are consistent with (or mirror) the validity principles of conditional logic (Forward Conditional and Contrapositon, respectively). Properties 2 and 3 are inconsistent with (or do not mirror) the fallacy principles of conditional logic (Inversion and Conversion, respectively). This is because Properties 2 and 3 allow us to draw definitive conclusions from the premises whereas we are not permitted to do so under the material conditional interpretation of conditional logic.

In a certain sense, we can consider the standard conditional branch statement as true (or valid) when "a" and "b" co-occur, or when "$\bar{a}$" and "$\bar{b}$" co-occur. In a similar sense, we can consider the standard conditional branch statement as false (or invalid) when "$\bar{a}$" and "b" co-occur or when "a" and "$\bar{b}$" co-occur. This view of the standard conditional branch statement mirrors the biconditional interpretation of the "if-then" conditional statement. The definitive conclusions drawn are the conclusions one is permitted to draw under the biconditional interpretation of the conditional statement. See the validity principles in Figure 1 and the biconditional "fallacy" principles in Figure 2.

If it is the case that transfer of learning from standard conditional branch statements to ordinary-language conditional statements occurs, improved performance on the validity principles of conditional logic is expected to occur independent of interpretation of ordinary-language conditional statements (i.e., material conditional or biconditional). This assertion (all other things being equal) is reflected in the following two informal hypotheses.

Informal Hypothesis 1. Learning the standard conditional branch statement might tend to improve the subject's performance on the validity principles of conditional logic under the material conditional interpretation.

Informal Hypothesis 2. Learning the standard conditional branch statement might tend to improve the subject's performance on the validity principles of conditional logic under the biconditional interpretation.

On the other hand, if it is the case that transfer of learning occurs and subjects interpret the ordinary-language conditional statement in a material conditional manner, then reduced performance is to be expected on the fallacy principles. This assertion is reflected in the following informal hypothesis.

<u>Informal Hypothesis 3.</u> Learning the standard conditional branch statement might tend to reduce the subject's performance on the fallacy principles of conditional logic under the material conditional interpretation.

Finally, if it is the case that transfer of learning occurs and subjects interpret the ordinary-language conditional statement in a biconditional manner, then improved performance is to be expected on the fallacy principles. This assertion is reflected in the following informal hypothesis.

<u>Informal Hypothesis 4.</u> Learning the standard conditional branch statement might tend to improve the subject's performance on the fallacy principles of conditional logic under the biconditional interpretation.

C. Explicit ELSE Conditional Branch Statements and Logical Principles

The standard conditional branch statement is but one type of conditional branch statement used in computer programming languages. Another is called the IF-THEN-ELSE conditional statement (or explicit ELSE) and it is used in many popular computer languages.[15] This statement has the general form:

IF a THEN b ELSE c (2)

where "a" is a predicate that evaluates to the truth-value of TRUE or FALSE. As with the standard conditional branch statement, we call "a" the antecedent-proposition. "b" and "c" are expressions, but are not propositions. Like the "b" in the standard conditional branch statement, they cause some action to be carried out by the program. I call "b" and "c," consequent-action-1 and consequent-action-2, respectively.

The explicit ELSE conditional branch statement has the logical structure shown in Figure 5.

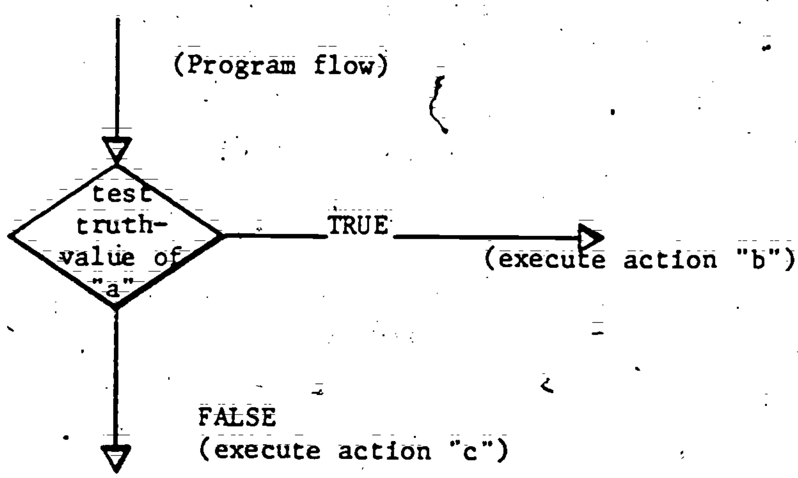

Figure 5. Logic of the Explicit ELSE Conditional Branch Statement

Like the standard conditional statement, there are two primary semantic properties associated with the explicit ELSE conditional branch statement:

1. When the antecedent-proposition is true ("a"), consequent-action-1 will <u>causally</u> be true ("b") <u>and</u> consequent-action-2 will <u>causally</u> be false ("$\bar{c}$").

2. When the antecedent-proposition is false ("$\bar{a}$"), consequent-action-1 will <u>causally</u> be false ("$\bar{b}$") <u>and</u> consequent-action-2 will <u>causally</u> be true ("c").

Four <u>derivative</u> semantic properties can be drawn from these primary semantic properties:

3. If we know that consequent-action-1 is true ("b"), then we are entitled to conclude that the antecedent-proposition is true ("a"). We are also entitled to conclude that consequent-action-2 is false ("$\bar{c}$").

4. If we know that consequent-action-1 is false ("$\bar{b}$"), then we are entitled to conclude that the antecedent-proposition is false ("$\bar{a}$"). We are also entitled to conclude that consequent-action-2 is true ("c").

5. If we know that consequent-action-2 is true ("c"), then we are entitled to conclude that the antecedent-proposition is false ("$\bar{a}$"). We are also entitled to conclude that consequent-action-1 is false ("$\bar{b}$").

6. If we know that consequent-action-2 is false ("$\bar{c}$"), then we are entitled to conclude that the antecedent-proposition is true ("a"). We are also entitled to conclude that consequent-action-1 is true ("b").

The co-occurrence combinations for the antecedent-proposition and both consequent-actions are shown in Table 2.

TABLE 2

CO-OCCURRENCE OF EXPLICIT ELSE ANTECEDENT-PROPOSITION AND CONSEQUENT-ACTIONS

Antecedent-Proposition	Consequent-Action-1	Consequent-Action-2	Co-occurrence
a	b	c	F
a	b	$\bar{c}$	T
a	$\bar{b}$	c	F
a	$\bar{b}$	$\bar{c}$	F
$\bar{a}$	b	c	F
$\bar{a}$	b	$\bar{c}$	F
$\bar{a}$	$\bar{b}$	c	T
$\bar{a}$	$\bar{b}$	$\bar{c}$	F

Note: "T" stands for co-occurrence and "F" stands for non-co-occurrence.

The explicit ELSE conditional branch statement is unlike the logical conditional "if-then" statement in two respects:

1. Like the standard conditional branch statement, the consequences are not propositions but can be considered to be quasi-propositional.

2. Each explicit ELSE conditional branch statement contains two consequent actions (or two quasi-consequences).

The first difference is not important providing that programmers do not make the kind of sophisticated distinction that we have made between propositions and quasi-propositions. The second difference warrants exploration.

1. Explicit ELSE Conditional Statements and Logical Validity Principles

The presence of two explicit ELSE consequent-actions (one affirmed whenever the other is denied) make the relationship between the explict ELSE conditional statement and the validity principles of conditional logic somewhat complex. The Forward Conditional principle permits us to affirm the (singular) consequent given the affirmation of the antecedent. In the explicit ELSE version of the Forward Conditional, we are entitled to affirm consequent-action-1 ("b") and at the same time to deny consequent-action-2 ("$\bar{c}$"), given that the antecedent proposition is affirmed ("a").

Similarly, the Contraposition principle allows us to deny the antecedent given that we deny the (singular) consequent. In the explicit ELSE version of Contraposition, the denial of consequent-action-1 ("$\bar{b}$") entitles us to conclude that the antecedent-proposition is denied ("$\bar{a}$"). On the other hand, the denial of consequent-action-2 ("$\bar{c}$") entitles us to affirm the antecedent-proposition ("a").

Thus, we can say that the explicit ELSE conditional branch statement is in some ways consistent with the conditional logic validity principles and in other ways inconsistent with these very same principles. This makes it, at best, confusing and I am led to conclude that the LOGO conditional branch statement is inconsistent (does not mirror or is incompatible) with the conditional logic validity principles under either the material conditional or the biconditional interpretation.

Learning explicit ELSE conditional branch statements might indirectly affect the programmer's ability to handle the "if-then" logical validity principles. This, as we shall see in the next section, may be contingent upon the programmer's interpretive model of the explicit ELSE conditional statement.

From the above, we can develop two informal hypotheses:

Informal Hypothesis 5. Learning explicit ELSE conditional branch statements might tend to reduce the subject's performance on the validity principles of conditional logic under the material conditional interpretation.

Informal Hypothesis 6. Learning explicit ELSE conditional branch statements might tend to reduce the subject's performance on the validity principles of conditional logic under the biconditional interpretation.

2. Explicit ELSE Conditional Statements and Logical Fallacy Principles

The fallacy principles of the logical "if-then" conditional, under the material conditional interpretation, do not permit us to draw a conclusion based upon the denial of the antecedent nor do they permit us to draw a conclusion based upon the affirmation of the (singular) consequent. On the other hand, the explicit ELSE conditional branch statement entitles us to draw a conclusion based upon these conditions. Thus, I am led to conclude that in this respect, the logic of the explicit ELSE conditional branch statement is inconsistent (does not mirror or is incompatible) with the two fallacy principles of conditional logic (Inversion and Conversion), under the material conditional interpretation.

However, when the conditional logic statement is interpreted in the biconditional manner, it is permissible to draw a conclusion based upon the denial of the (singular) consequence. See Figure 2. This fact should not tempt us to jump to the conclusion that the explicit ELSE conditional branch statement is inconsistent with the logical conditional statement under the material conditional interpretation but consistent under the biconditional interpretation. Clearly, the denial of the explicit ELSE antecedent-proposition entitles us to affirm one consequent-action. In addition, the affirmation of one of the explicit ELSE consequent-actions (b) permits the affirmation of the explicit ELSE antecedent-proposition, but affirmation of the other consequent-action (c) permits the denial of the same antecedent-proposition.

At best this makes matters confusing and at worst I am led to conclude, as I have for the logical validity principles, that the explicit ELSE conditional branch statement is inconsistent with the logical conditional under the biconditional interpretation.

Learning explicit ELSE conditional branch statements might affect the programmer's ability to handle the "if-then" logical fallacy principles. From the above, we can develop two informal hypotheses.

Informal Hypothesis 7. Learning explicit ELSE conditional branch statements might tend to reduce the subject's performance on the fallacy principles of conditional logic under the material conditional interpretation.

Informal Hypothesis 8. Learning explicit ELSE conditional branch statements might tend to reduce the subject's performance on the fallacy principles of conditional logic under the biconditional interpretation.

The analysis of the explicit ELSE conditional branch statement that leads to the above informal hypotheses reflects one of two distinct ways to interpret these conditionals. I call the above interpretation the explicit ELSE interpretation to distinguish it from the implicit ELSE interpretation described in the next section.

D. Implict ELSE Interpretation and Logical Principles

Consider again the standard conditional branch statement:

IF a THEN b (1)

This statement can be interpreted as containing an implied ELSE clause:

IF a THEN b [ELSE c] (3)

where "c" stands for "go to the next sequential program statement." The brackets serve to indicate that the ELSE clause is concealed or implied.

There is another way to render (3). It is a way that better reflects the implicitness of the ELSE clause:

IF a THEN b [ELSE $\bar{b}$], (4)

This rendering means that if "a" is false ("$\bar{a}$"), do not carry out consequent-action "b" ("$\bar{b}$").

Consider the explicit ELSE conditional branch statement repeated below:

IF a THEN b ELSE c. (2)

If it is the case that the subject focuses upon the first consequent-action ("b") when dealing with these conditional branch statements, then although "ELSE c" is explicitly present in the statement, this clause might be interpreted in a secondary or implicit manner. This possible interpretation is illustrated in (4).

If the ELSE clause is interpreted in this manner, then the logic of the explicit ELSE conditional statement reduces to the standard conditional statement logic (see the four standard conditional branch statement properties). This interpretation was suggested by the way in which subjects sometimes explained the operation of the LOGO conditional branch statement to teachers during the study reported here.[16] Under this implicit interpretation of the LOGO conditional statement, the semantics reflect and are consistent with the biconditional interpretation of the logical "if-then" conditional statement and as such are thus consistent with the validity principles and inconsistent with the fallacy principles of conditional logic, under the material conditional interpretation.

Learning LOGO conditional branch statements under an implicit ELSE interpretation, might affect the child's ability to handle the "if-then" logical principles. From the above, we can see that the informal hypotheses that could be generated here would be identical to Informal Hypotheses 1 through 4 for the standard conditional branch statement.[17]

With this analysis complete, I now present an experiment that tests the hypotheses that have been developed.

III

THE EXPERIMENT AND RESULTS

A. The Design

The computer programming language chosen for this study is LOGO, a language which utilizes the IF-THEN-ELSE conditional branch statement. LOGO is a LISP-like[18] computer programming language with an English-like syntax that makes it fairly easy to learn. In fact, LOGO was developed for young children to use as a framework within which to learn mathematics. Some claim that it can help children learn just about any formal subject.[19]

This study makes no attempt to directly determine which IF-THEN-ELSE interpretation (explict or implicit ELSE) holds for subjects nor is there any attempt to determine whether subjects interpret the logical conditional statement in a material conditional or biconditional manner. Determinations such as these are fraught with methodological difficulties.[20]

To try to answer the central research question a post-test only control group design was utilized. It is one of the three "true" designs described by Campbell and Stanley (1963).[21] Subjects in the fifth-grade of a public elementary school were randomly selected for the experiment and then randomly placed in either the experimental or control group. Fifth-grade students (10-11 years of age) were chosen since previous attempts to teach LOGO programming to this age group had been highly successful (Papert, 1972a; Statz, 1973) and the empirical data on this age group's ability to utilize correctly the principles of conditional logic was extensive.[22]

The experimental group was taught the LOGO computer programming language. The control group was not taught LOGO and received no special instruction of any kind. At the conclusion of the treatment period, both groups were given a series of tests measuring their conditional reasoning abilities. These tests were scored in <u>two</u> distinct ways: 1) using the material conditional interpretation as the standard of correctness of test answers and 2) using the biconditional interpretation as the standard of correctness of test answers. See Figures 1 and 2. Clearly, scoring the measures under only one interpretation of the conditional statement might result in the loss of important data. The experimental and control groups were statistically compared within both scoring schemes by split-plot two-factor repeated measures and one-way analysis of variance statistical analyses.

B. <u>Subjects and Their Environment</u>

This study was carried out in an elementary school serving a racially mixed suburban neighborhood in Syracuse, New York. The school was fully integrated having about equal numbers of whites and non-whites in each grade. The children came from families that spanned the socioeconomic spectrum.

Fifth graders were chosen for this study because their age group represented the upper-end of the concrete operational stage of cognitive development, close to the transition period into formal operations (Inhelder and Piaget, 1958). All fifth grade teachers were highly cooperative with the experimenter and 42 fifth graders were chosen at random for the experiment. Of the 42, 21 were randomly assigned to the experimental group (EG) and the other 21 were assigned to the control group (CG). One subject dropped out of the EG midway through the experiment (the family moved out of town) and was not replaced.

The following data were obtained for each child in the study:

1. chronological age
2. sex
3. standardized achievement scores (California Achievement Test)
4. scores on the Smith-Sturgeon Conditional Reasoning Test (Ennis, 1969)
5. scores on an abridged version of The Cornell Conditional Reasoning Test, Form X (Ennis and Paulus, 1965)
6. scores on the LOGO Conditional Branch Test, for EG only (Seidman, 1980a).

The first three items were obtained from school records and are reported in Tables 3 and 4. IQ scores were not available as a matter of School District policy. The remaining data items were obtained at the end of the treatment period (except for achievement pre-tests which were administered prior to the experimental treatment).

TABLE 3

MALES, FEMALES, CHRONOLOGICAL AGE AND TOTALS

	Males		Females		Total	
	N	CA	N	CA	N	CA
EG	7	139.14 (6.87)	13	135.00 (6.99)	20	136.45 (7.06)
CG	12	139.75 (15.06)	9	140.78 (10.07)	21	140.19 (12.87)

Note: Number (N), mean chronological age in months (CA) and standard deviation of CA (in parentheses) of males and females in the experimental (EG) and control (CG) groups.

Pre and post-test (given immediately before and after the LOGO experience) achievement scores were available for both groups of subjects. A one-way analysis of variance between the experimental and control groups on all six pre-test achievement scores found no statistically significant (hereafter, "significant") difference between the two groups. Thus, the random selection and assignment of subjects seems to have assured no significant difference between the two groups on these achievement measures.

Similarly, a one-way analysis of variance between the two groups on post-test achievement scores found no significant differences. This seems to indicate that the experimental treatment (learning to program in LOGO) did not significantly differentiate the two groups along these achievement measures.

TABLE 4

PRE AND POST-TEST ACHIEVEMENT SCORES FOR EXPERIMENTAL AND CONTROL GROUPS
CALIFORNIA ACHIEVEMENT TEST

PRE-TEST						POST-TEST					
READING			MATH			READING			MATH		
Vocab-ulary	Compre-hension	Total	Compu-tation	Con-crete	Total	Vocab-ulary	Compre-hension	Total	Compu-tation	Con-crete	Total
28.72 (7.19)	25.0 (7.5)	53.72 (14.05)	38.41 (12.5)	25.83 (8.43)	64.0 (20.09)	27.25 (10.28)	25.3 (9.2)	52.5 (18.6)	41.2 (12.23)	30.75 (6.39)	71.95 (17.93)
29.1 (6.76)	21.10 (7.82)	49.89 (12.63)	40.58 (9.84)	29.68 (4.74)	70.26 (13.87)	29.9 (6.24)	26.10 (8.42)	56.0 (12.9)	46.47 (9.0)	32.21 (4.6)	78.68 (12.39)

Mean achievement raw scores and standard deviations (in parentheses). Missing values are excluded from the tics.

A comparison of the pre and post-test achievement scores for the EG was performed by a repeated measures test (Morrison, 1967, 133-141). Significant differences, favoring the post-test, were found for the math computation subscore ($F=14.26$, $df=1, 17$) and for the math total score ($F=10.8$, $df=1, 16$). This same analysis performed on the control group found significant differences, favoring the post-test, for reading total scores ($F=4.65$, $df=1, 18$), math computation subscore ($F=5.34$, $df=1, 16$) and math total score ($F=7.05$, $df=1, 16$).

This analysis shows that the CG improved significantly in their total reading score but on all other achievement measures were identical to the experimental group.

C. Educational Treatment

Four portable teletypewriter computer terminals connected by telephone lines to the Syracuse University PDP-10 computer were situated, along with the LOGO Turtle, in a section of the mathematics laboratory in the school.[23] The mathematics laboratory was a separate room that served the school as a resource center for mathematics skill building and enrichment. Children came to do LOGO work in groups of fours and on a scheduled basis: one hour sessions, twice each week for a total of fifteen weeks. This meant that each child in the EG received 30 hours of LOGO instruction and experience (all missed periods were made up). Each child had exclusive access to a time-shared teletypewriter and the children in each LOGO group took turns using the Turtle. There were always at least two LOGO teachers present at each of the sessions.

LOGO programming was taught to the EG using a guided problem centered approach modeled somewhat after Papert's work with children (Papert, 1972b). After some very basic instruction in the use of the teletypewriter and Turtle, the children were encouraged to generate their own projects and problems.

The LOGO teachers had a range of problem types that they suggested to the children so that they would encounter and learn various LOGO programming concepts. All teachers tried to incorporate a series of LOGO concepts into the children's project experiences. Figure 6 contains a list of these concepts.

1. familiarization with machinery, typing, signing on and off
2. procedure form - TO, END, and use of PRINT
3. teletype designs using PRINT
4. simple Turtle procedures using the Turtle commands: FORWARD, BACK, RIGHT, LEFT, PENUP, PENDOWN
5. use of subprocedures with teletype and Turtle
6. use of open-ended recursion (i.e., no limits imposed to stop the recursion)
7. introduce NAME, THING and the use of inputs
8. introduce SENTENCE, SENTENCES, WORD and WORDS
9. introduce RANDOM
10. introduce TYPEIN and some interactive games
11. introduce the conditional statement: IF-THEN-ELSE
12. use of full recursion with inputs
13. use of stop rules and limits
14. use of recursion, OUTPUT and combinations of procedures using OUTPUT

Figure 6. List of LOGO Concepts and Approximate Order Introduced (Adapted from Statz, 1973, Figure 11)

Projects and problems chosen by the children with teacher guidance generally broke down into three categories: simple projects, projects utilizing variables, and projects utilizing decision points. See Figure 7 for some examples provided by Statz (1973). No order of encounter is implied by the list. LOGO teachers (with extensive training) guided the children through these project types making sure that the concepts listed in Figure 6 were taught. Although the children did much of their LOGO work alone, they were encouraged to help one another and as a result, a convivial and cooperative atmosphere pervaded the sessions.

Simple Projects

1. Teletype Projects - designs of initials, animals, figures to be printed, combinations for complex figures, "simple recursion"
2. Turtle Projects - designs done with strings of turtle commands, complex figures using several subprocedures of simple designs, "simple recursive" circles and near-circles
3. New Concepts - procedure, subprocedure, editing, recursion, filing
4. New LOGO Commands and Operations - TO, END, EDIT, SAVE, GET, LIST, ERASE, Turtle commands

Projects with Variables

1. Teletype Projects - procedures with variables as inputs to arithmetic or language games
2. Turtle Projects - expandable figures
3. New Concepts - inputs, variable names and values
4. New LOGO Commands and Operations - MAKE, TYPEIN, string operations

Projects with Decision Points

1. Teletype Projects - games with limits and conditional branches
2. Turtle Projects - designs built on arcs
3. New Concepts - limit, stop rule, conditional
4. New LOGO Commands and Operations - TEST, IFTRUE, IFFALSE, GO TO LINE, relational operators, [IF-THEN-ELSE]

Figure 7. Categories of LOGO Projects
(Statz, 1973, Figure 12)

A central focus of the LOGO experience for the children was learning to use the LOGO conditional branch statement within their LOGO procedures. The experimental group was taught the "IF-THEN-ELSE" form of the LOGO conditional branch statement. The children were usually exposed to the LOGO conditional branch when they needed to execute different parts of their LOGO procedures depending upon the value of a particular LOGO variable. These kinds of situations arose often in game playing and question asking/answering procedures. Children also needed to acquire the concept of LOGO conditional branch when limit points in iterative and recursive procedures were needed.

Four stages of acquiring LOGO conditional branch statement syntax and semantics have been identified by Statz (1973). They are:

1. Introduction Stage. The need for conditional branches usually crops up in the context of a problem situation that the child is working on. For example, if the child is writing a number guessing procedure that compares a randomly generated secret number against a user's guess, the teacher might illustrate a solution such as this:

```
>10 IF :GUESS = :SECRET THEN PRINT "RIGHT" ELSE PRINT "WRONG"
```

A child might wish, for example, to print the integers between "x" and "y," inclusive. The teacher might demonstrate this solution:

```
>?TO COUNTUP :X :Y
>10 PRINT :X
>20 "X"<— (:X+1)
>30 IF :X > :Y THEN STOP ELSE GOTO LINE 10
>40 END
```

It is in this stage that the semantics and syntax of the LOGO conditional branch statement are illustrated and introduced to the student.

2. Structure and Re-explanation Stage. Sometimes children need help in determining just where in their procedures to place the conditional branch. These are structural problems and are not directly related to the syntax and semantics of the conditional itself. On the other hand, children sometimes need a re-explanation of the general aspects of the syntax and semantics of the LOGO conditional statement. Misplaced and fragmented conditionals were found to be frequent at this stage of learning the LOGO conditional branch statement.

3. Reminding Stage. Here, children need only simple reminders about the syntax and semantics of the LOGO conditional branch statement, rather than re-explanations. Often, children compared the wrong variables in the logical expression and sometimes they would mix up the placement of the THEN and ELSE clauses. Simple reminders were found sufficient to correct these conditions.

4. Fluency Stage. At this stage, children easily placed the conditionals correctly within their procedures and understood the syntax and semantics to the point where very few errors occured.

Although LOGO teachers were especially concerned that all children reach this last stage, not all of the subjects in the EG achieved the Fluency Stage (see next section).

D. Measures and Results

1. LOGO Conditional Branch Statement

The LOGO Conditional Branch Test was devised to test how well the EG understood the semantics of the LOGO conditional branch statement. The questions and tester script for the LOGO Conditional Branch Test can be found in Seidman (1980a). The test measured four aspects of the LOGO conditional branch statement logic. These aspects, in their general form, are shown in Figure 8.

1. Affirm Antecedent: Given that "a" is affirmed, is it true that "b" occurs?

2. Deny Consequent: Given that "a" is denied, is it true that "c" occurs?

3. Consequent-Action "b" Occurs: Given that "b" occurs, is it true that "a" is affirmed?

4. Consequent-Action "c" Occurs: Given that "c" occurs, is it true that "a" is denied?

Figure 8. - Four Aspects of the LOGO Conditional Branch Statement Tested by LOGO Conditional Branch Test (The General form of the LOGO conditional statement is IF a THEN b ELSE c)

In the test, the antecedent-proposition, "a," always consisted of two integers connected by an equal sign. Consequent-actions, "b" and "c," always consisted of a PRINT command that output a very simple and familiar word (a LOGO literal). Questions 1 and 2 in the test were designed to determine whether or not the subject knew that "a" was true when two identical numbers were connected by the equal sign. Questions 3 and 4 in the test are designed to determine whether the subject correctly understood the workings of the PRINT command. All of the subjects in the EG correctly answered test Questions 1 through 4.

The remaining twenty-four questions in the test were made up of 6 questions for each of the four aspects of the LOGO conditional branch statement. These questions were randomly arranged with the proviso that no two questions in any one category could follow one another. Each question in the test had four answers to choose from. The last two answers were always:

c.) can't tell from the information given
d.) don't know.

The correct answer and an incorrect answer were randomly assigned to the first and second answer choices. Scoring was quite stringent. A subject was given a correct score on a question only if he or she chose the correct answer from amongst the four choices and gave the correct reason for answering as he or she did. Example questions and correct sample reasons for each of the aspects of the LOGO conditional branch statement are shown in Figure 9. The tester pretends, with the subject, to type a LOGO conditional branch statement on the teletypewriter. The subject and the tester also pretend that the "terminal" responds. The "XXXX" box in the examples indicate the part of the "input" conditional statement or the "terminal" response that is covered up for the purpose of the test.

Table 5 shows the means and standard deviations of the EG scores on each of the four aspects of the LOGO Conditional-Branch Test and the total scores. In addition, the table shows what percentage of the subjects obtained 5 or 6 correct answers (out of 6 for each aspect), 4 correct answers and less than 4 correct answers.

1. Affirming the Antecedent

```
?IF 18 = 18 THEN PRINT "HORSE" ELSE PRINT "COW"
XXXX
                         Answers
                         a. COW
                         b. HORSE
                         c. can't tell from the information given
                         d. don't know
```

Sample Correct Reason for Correct Answer, "HORSE": "Since 18 is equal to 18, HORSE gets printed" or "HORSE is printed because 18 is equal to 18."

2. Denying the Antecedent

```
?IF 15 = 19 THEN PRINT "HEAD" ELSE PRINT "FOOT"
XXXXX
                         Answers
                         a. HEAD
                         b. FOOT
                         c. can't tell from the information given
                         d. don't know
```

Sample Correct Reason for Correct Answer, "FOOT": "15 is not equal to 19, so FOOT is printed" or "FOOT is printed because 15 and 19 are not the same."

3. Consequent-Action "b" Occurs

```
?IF  XXXXX THEN PRINT "ROOM" ELSE PRINT "PENCIL"
ROOM
                         Answers
                         a. 30=30
                         b. 37=35
                         c. can't tell from the information given
                         d. don't know
```

Sample Correct Reason for Correct Answer, "30=30": "ROOM" had to be printed because 30 equals 30 must have been under the box" or "it must have been 30=30 under the box because ROOM was printed."

4. Consequent-Action "c" Occurs

```
?IF  XXXXX THEN PRINT "EAR" ELSE PRINT "NOSE"
NOSE
                         Answers
                         a. 52=57
                         b. 55=55
                         c. can't tell from the information given
                         d. don't know
```

Sample Correct Reason for Correct Answer, "52=57": "NOSE was printed because 52=57 was after the IF" or "It must have been 52=57 because NOSE was printed."

Figure 9. Sample Questions — LOGO Conditional Branch Test and Sample Correct Reasons for Correct Answers

TABLE 5

TOTAL SCORES AND SUBSCORES ON LOGO CONDITIONAL BRANCH TEST

	Affirm Antecedent	Deny Antecedent	Consequent Action "b" occurs	Consequent Action "c" occurs	Total
Mean	5.1	3.95	4.25	4.1	17.4
Standard Deviation	0.45	0.22	0.639	0.447	1.07
5 or 6 Correct	95	0	25	15	
4 Correct	5	95	70	80	
Less Than 4 Correct	0	5	5	5	

In each aspect of the LOGO conditional branch statement tested for, 95% or more of the experimental group answered 4 or more of the questions correctly. If we use the Ennis criteria for mastery of logical conditional statements (Ennis and Paulus, 1965), we can say that for each aspect of the LOGO conditional branch statement, 95% of the experimental group satisfies the necessary condition of mastery.

In addition, if we define LOGO conditional branch statement Fluency as meeting the sufficient condition for mastery (5 or 6 out of 6 correct), then we can say that most of the experimental subjects were fluent on affirming the antecedent but very few were fluent on the other three LOGO conditional branch aspects. However, if we define being in the Reminder Stage as meeting the necessary condition for mastery (at least 4 out of 6 correct), then we can say that at least 95% of the experimental subjects were at this stage for each of the four LOGO conditional branch statement aspects.

2. Principles of Conditional Logic

There are a number of ways researchers have gone about measuring performance on the principles of conditional logic.[24] In this study two distinct measures devised by Ennis and his associates were used: an abridged version of the Cornell Conditional Reasoning Test, Form X (CCRT) and the Smith-Sturgeon Conditional Reasoning Test (SSCRT). The CCRT (Ennis and Paulus, 1965) is a paper and pencil test and the SSCRT (Ennis, 1969) is a concrete manipulation test, and were both devised to measure the same principles of conditional reasoning. The two tests are complementary in that the SSCRT was created to avoid the dependence upon reading skills that is characteristic of the CCRT. To my knowledge, these two tests have never been used in the same experiment to measure principles of conditional logic.

The test results were scored using both a material conditional interpretation and a biconditional interpretation of the logical conditional statement. The only possible differences between the two scoring schemes can appear in the fallacy principles.[25]

a. Cornell Conditional Reasoning Test

An abridged version of the Cornell Conditional Reasoning Test, Form X was used in this study to measure both the control and experimental group's understanding of the four principles of conditional logic (see Figure 1). The original test was cut in half for this purpose, as Ennis and Paulus (1965) suggest, and the questions used were randomly mixed with the proviso that no two questions from any one item group appear on the same page. The two item groups of questions pertaining to Transitivity are excluded from the data analysis reported here.

Because of the split-plot repeated measures post-test-only control group design utilized in this study, subjects took this test at different times. However, whenever the test was taken, it was taken at one sitting.

Table 6 shows the means and standard deviations of the scores on the four principles of conditional logic under both interpretations for the EG. In addition, the table shows the percentage of the subjects in the EG who correctly answered 5 or 6 questons, 4 questions and less than 4 questions for the various principles. Table 7 presents the same data for the CG.

TABLE 6

EXPERIMENTAL GROUP MEANS AND MASTERY DATA ON FOUR PRINCIPLES OF CONDITIONAL LOGIC MEASURED BY THE CCRT

	Forward Conditional	Inversion		Conversion		Contra-Positional
	MC/BC	MC	BC	MC	BC	MC/BC
Mean	3.1	1.65	3.8	1.65	3.85	2.2
SD	1.41	1.49	1.9	1.30	1.69	1.54
5 or 6 Correct (%)	15	0	40	0	40	10
4 Correct (%)	20	20	15	10	15	10
Less than 4 correct (%)	65	80	45	90	4	80

Note: "MC" stands for material conditional interpretation of answers; "BC" stands for biconditional interpretation of answers; "MC/BC" indicates that both interpretations give the same results.

TABLE 7

CONTROL GROUP MEANS AND MASTERY DATA ON FOUR PRINCIPLES OF CONDITIONAL LOGIC MEASURED BY THE CCRT

	Forward Conditional	Inversion		Conversion		Contra-Positional
	MC/BC	MC	BC	MC	BC	MC/BC
Mean	3.19	1.80	2.19	1.66	3.33	2.33
SD	1.43	1.47	1.59	1.27	1.45	1.51
5 or 6 Correct (%)	19	0	10	0	19	9.5
4 Correct (%)	19	19	14	10	19	9.5
Less than 4 correct (%)	62	81	76	90	62	81

Note: "MC" stands for material conditional interpretation of answers; "BC" stands for biconditional interpretation of answers; "MC/BC" indicates that both interpretations give the same results.

Notice how dramatically better the EG does on the fallacy principles under the biconditional interpretation. For Inversion, a total of 55% satisfy the necessary condition for mastery compared to 20% under the material conditional interpretation. For Conversion, 55% satisfy the necessary condition for mastery compared to 10% under the material conditional interpretation. The results for the CG under the biconditional interpretation for the fallacy principles are less dramatic. For Inversion, 24% satisfy the necessary condition for mastery compared to 19% under the material conditional interpretation. For Conversion, 38% satisfy the necessary condition for mastery compared to 10% under the material conditional interpretation.

b. Smith-Sturgeon Conditional Reasoning Test

The other test used in this study to measure the ability to utilize correctly principles of conditional logic was the SSCRT. This test consists of two parts: a <u>house</u> part and a <u>chemical</u> part. Each part was separately administered in a random fashion to each child in the study. For both parts of the test, the tester must first determine whether or not the subject understands the Forward Conditional principle before the subject is allowed to proceed with the test. All subjects understood this principle and all were allowed to proceed with the test. Both parts of the test measured Inversion, Conversion, Contraposition and Transitivity. Transitivity scores were excluded from the data analysis reported in this study.

Table 8 shows the mean and standard deviation scores for the EG on each of the three principles as well as the percentage of the group that answered 5 or 6 questions correctly, 4 questions correctly and less than 4 questions correctly for each principle. These data are presented for both interpretation of the test question answers. Table 9 presents the same data for the CG.

Notice that for both the experimental and control groups, a sizable proportion does not meet the necessary condition for mastery. For both groups, considerably more subjects meet the necessary condition for mastery under the biconditional interpretation than under the material conditional interpretation.

TABLE 8

EXPERIMENTAL GROUP MEANS AND MASTERY DATA ON THREE PRINCIPLES OF CONDITIONAL LOGIC MEASURED BY THE SSCRT

		House	Chemical	Total	5-6 Correct (%)	4 Correct (%)	Less than 4 Correct (%)
Inversion	MC	0.15 (0.49)	2.15 (1.53)	2.3 (1.66)	5	20	75
	BC	1.75 (0.55)	1.10 (0.97)	2.85 (1.27)	5	30	65
Conversion	MC	0.40 (0.60)	2.15 (1.42)	2.55 (1.73)	15	10	75
	BC	1.35 (0.745)	1.50 (1.28)	2.85 (1.7)	15	25	60
Contra-position	MC/BC	0.65 (0.875)	3.5 (0.88)	4.15 (1.18)	35	50	15

Note: "MC" stands for material conditional interpretation of answers; "BC" stands for biconditional interpretation of answers; "MC/BC" indicates that both interpretations give the same results.

TABLE 9

CONTROL GROUP MEANS AND MASTERY DATA ON THREE PRINCIPLES OF CONDITIONAL LOGIC MEASURED BY THE SSCRT

		House	Chemical	Total	5-6 Correct	4 Correct	Less than 4 Correct
Inversion	MC	0.05 (0.22)	2.33 (1.24)	2.38 (1.32)	5	19	76
	BC	1.81 (0.40)	1.14 (0.96)	2.95 (0.97)	5	24	71
Conversion	MC	0.48 (0.81)	1.71 (1.42)	2.19 (1.83)	14	5	81
	BC	1.43 (0.87)	2.05 (1.32)	3.48 (1.75)	24	38	38
Contra-position	MC/BC	0.57 (0.81)	2.81 (1.25)	3.38 (1.63)	19	33	48

Note: "MC" stands for material conditional interpretation of answers; "BC" stands for biconditional interpretation of answers; "MC/BC" indicates that both interpretations give the same results.

c. Correlations: LOGO Conditional Branch Statement and Principles of Conditional Logic

Table 10 shows the Pearson Product Moment correlations between EG scores on the four aspects (and totals) of the LOGO conditional branch statements as measured by the LOGO Conditional Branch Test and the four principles of conditional logic as measured by the CCRT and SSCRT under both the material-conditional and biconditional interpretations.

TABLE 10

CORRELATIONS BETWEEN PRINCIPLES OF CONDITIONAL LOGIC AND LOGO CONDITIONAL BRANCH STATEMENT ASPECTS

			LOGO Conditional Statement Test				
			Affirming "a"	Denying "a"	Affirming "b"	Affirming "c"	Total
Forward Conditional	CCRT	MC/BC	-.1001	-.4839*	-.0876	-.1835	-.265
Inversion	CCRT	MC	.2595	.2909	.2060	-.1330	.231
		BC	.08599	.4620*	-.1702	-.1580	-.035
	SST-H	MC	.1683	.0722	-.1263	-.0722	-.019
		BC	-.0170	-.0169	-.1124	.1070	-.087
	SST-C	MC	-.1768	.1768	.0135	.2843	.0878
		BC	.0972	-.2188	-.2128	-.0243	-.139
	SST-T	MC	-.1136	.1846	-.0248	.241	.075
		BC	.0278	-.2134	-.2112	.0278	-.143
Conversion	CCRT	MC	.1528	.2967	.2361	.2428	.359
		BC	.1597	.2509	-.1580	-.3959	-.136
	SST-H	MC	.2361	.1574	0	.0393	.1445
		BC	-.4264	-.2053	-.0830	.2053	-.180
	SST-C	MC	.0578	.3552	-.1012	.2230	.128
		BC	-.0921	-.4606*	-.0968	-.1842	-.263
	SST-T	MC	.1291	.3466	-.0832	.1971	.155
		BC	-.2569	-.4375	-.1094	-.0486	-.277
Contra-Position	CCRT	MC/BC	-.335	-.4270	-.2671	-.2594	-.4859*
	SST-H	MC/BC	-.0404	.1748	-.1177	.0941	-.010
	SST-C	MC/BC	-.1325	.1325	-.1391	.5298*	.1087

Note: "MC" stands for material conditional interpretation of answers; "BC" stands for biconditional interpretation of answers; "MC/BC" indicates that both interpretations give the same results. "CCRT" = Cornell Conditional Reasoning Test, Form X. "SST" = Smith-Sturgeon Conditional Reasoning Test. "-H" = House Part. "-C" = Chemical Part. "-T" = Total. * = statistically different from zero.

Note that in Table 10, the only correlation total that we can say is significantly different from zero is the one with the CCRT Contraposition principle and that one is negative. All other correlations between logical principles and total LOGO Conditional Statement Test scores are not significantly different from zero. The critical value, at the 0.05 level, for the Pearson Correlation is 0.444, df=18. See Roscoe (1969, Table A-11).

E. Hypotheses and Inferential Experimental Results

We want to determine whether or not the EG was affected by any indirect instruction in the four principles of conditional logic, as a result of the LOGO experience. Null hypotheses are developed for the validity and fallacy principles. In all of the inferential statistics reported in this section, a separate analysis was made with the pre- and post-test achievement scores statistically partialled out (analysis of covariance). Since there was no effect due to the partialling, the non-partialled statistical results are reported here. Results at the 0.05 significance level are considered to be statistically significant.

1. Validity Principles

Hypothesis 1. There is no significant difference between the experimental and control groups on the Forward Conditional principle when the test measuring performance on this principle is scored under the material conditional/biconditional interpretation.

The CCRT was used as the sole measure of this principle. Recall that the material conditional and biconditional interpretations (and thus the scoring) of the question answers are identical for validity principles. Table 11 shows the results of the one-way analysis of variance that tests Hypothesis 1.

There is no significant difference between the experimental and control groups on the Forward Conditional principle under the material conditional/biconditional interpretation as measured by the CCRT. Thus, Hypothesis 1 cannot be rejected.

TABLE 11

ONE-WAY ANALYSIS OF VARIANCE TEST FOR HYPOTHESIS 1

Source	SS	df	MS	F
Between Groups	0.0838	1	0.0838	0.041
Within Groups	79.04	39	2.027	---
Total	79.12	40		

<u>Hypothesis 2.</u> There is no significant difference between the experimental and control groups on the Contraposition principle when tests measuring performance on this principle are scored under a material conditional/biconditional interpretation.

The CCRT and the SSCRT are two separate measures involved in testing this hypothesis. These two tests were administered in a random order to all subjects. The tests were given a number of different times and subjects were randomly chosen to take them. It was thus possible to test this hypothesis using a split-plot repeated measures statistical design. This design is described in detail in Kirk (1968). Kodrof and Roberge (1975) utilize this design in their experiment which used a concrete materials and a verbal form of a test to measure two principles of conditional logic. Table 12 shows the results for Hypothesis 2.

Table 12 shows a significant main effect of measures. An examination of Tables 6, 7 and 9 shows that both groups did significantly better on the SSCRT than on the CCRT when measured on the Contraposition principle. There were, however, no significant main effects for groups and measures. Thus, there is no significant difference between the experimental and control groups on the Contraposition principle under the material conditional/biconditional interpretations, as measured by the CCRT and the SSCRT. Thus, Hypothesis 2 cannot be rejected.

TABLE 12

SPLIT-PLOT REPEATED MEASURES TEST OF HYPOTHESIS 2

Source	SS	df	MS	F	p
Between Subjects	79.488	40	—	—	—
Groups	2.737	1	2.737	1.39	0.244
$Error_b$	76.751	39	1.968	—	—
Within Groups	145.500	41	—	—	—
Measures	48.402	1	48.402	20.133	0.0002*
Groups X Measures	3.337	1	3.337	1.388	0.2444
$Error_w$	93.761	39	2.404	—	—
Total	224.988	81	—	—	—

Note: * indicates significance at the .05 level.

2. Fallacy Principles

Hypothesis 3a. There is no significant difference between the experimental and control groups on the Inversion principle when the tests measuring performance on this principle are scored under the material conditional interpretation.

Hypothesis 3b. There is no significant difference between the experimental and control groups on the Inversion principle when the tests measuring performance on this principle are scored under the biconditional interpretation.

A split-plot repeated measures statistical design, similar to the one that was performed to test Hypothesis 2, was utilized here. Table 13 shows the results for Hypothesis 3a and Table 14 shows the results for Hypothesis 3b.

Table 13 shows no significant results. There is no significant difference between the two groups on the Inversion principle under the material conditional interpretation as measured by the CCRT and SSCRT. Thus, Hypothesis 3a cannot be rejected. Neither is there any significant difference between the two measures used to test performance on this logical principle, nor is there any interaction effect between the groups and tests under the material conditional interpretation.

Table 14 shows significant groups and interaction (groupsXmeasures) effects. The simple main effects analysis, presented in Table 15 along with the means presented in Tables 6, 7, 8 and 9, shows that the experimental group did significantly better than the control group on the CCRT Inversion items. Thus, Hypothesis 3b can be rejected.[26]

In addition, Tables 6, 7, 8 and 9 indicate that the experimental group did significantly better on the CCRT Inversion items than they did on the SSCRT Inversion items. And, the control group did significantly better on the SSCRT Inversion items than they did on the CCRT Inversion items.

<u>Hypothesis 4a.</u> There is no significant difference between the experimental and control groups on the Conversion principle when tests measuring performance on this principle are scored under the material conditional interpretation.

<u>Hypothesis 4b.</u> There is no significant difference between the experimental and control groups on the Conversion principle when the tests measuring performance on this principle are scored under the biconditional interpretation.

A split-plot repeated measures statistical design, similar to the one that was performed to test Hypothesis 2, was utilized here. Table 16 shows the results for Hypothesis 4a and Table 17 shows the results for Hypothesis 4b.

Tables 16 and 17 show no significant results. Thus, Hypothesis 4a and Hypothesis 4b cannot be rejected. There is no significant difference between the two groups on the Conversion principle under the material conditional interpretation and under the biconditional interpretation as measured by the CCRT and SSCRT. Neither is there any significant difference between the two measures used to test this logical principle under either the material conditional or the biconditional interpretations. Finally, there is no interaction effect between the groups and the tests used to measure performance on the Conversion principle under the material conditional and biconditional interpretations.

TABLE 13

SPLIT-PLOT REPEATED MEASURES TEST OF HYPOTHESIS 3a

Source	SS	df	MS	F	p
Between Subjects	66.902	40	--	--	--
Groups	0.008	1	0.008	0.0049	0.9429
$Error_b$	66.894	39	1.715	--	--
Within Subjects	111.500	41	--	--	--
Measures	5.378	1	5.378	1.977	0.1642
Groups X Measures	0.0755	1	0.0755	0.0277	0.8628
$Error_w$	106.046	39	2.719	--	--
Total	178.402	81	--	--	--

TABLE 14

SPLIT-PLOT REPEATED MEASURES TEST OF HYPOTHESIS 3b

Source	SS	df	MS	F	p
Between Subjects	89.195	40	--	--	--
Groups	11.634	1	11.634	5.85	0.0192*
$Error_b$	77.561	39	1.988	--	--
Within Subjects	111.5000	41	--	--	--
Measures	0.1097	1	0.10976	0.0444	0.8285
Groups X Measures	15.010	1	15.010	6.0739	0.0173*
$Error_w$	96.3797	39	2.471	--	--
Total	200.695	81	--	--	--

Note: * indicates a significant result at the 0.05 level.

TABLE 15

ANALYSIS OF SIMPLE MAIN EFFECTS IN TABLE 42

Source	SS	df	MS	F
Between Subjects				
Between Groups at CCRT	26.53	1	26.53	927.6*
Between Groups at SSCRT	0.107	1	0.107	3.74
Within Cell	2.23	78	0.0286	
Within Subjects				
Between Measures at EG	336	1	336	136*
Between Measures at CG	218	1	218	88*
AB	15.01	1	15.01	6.073
Meas. X Subjects with Groups	96.379	39	2.47	
Total	200.695	79		

Note: * indicates significance at the 0.05 level.

TABLE 16

SPLIT-PLOT REPEATED MEASURES TEST OF HYPOTHESIS 4a

Source	SS	df	MS	F	p
Between Subjects	77.49	40	—	—	—
Groups	0.60	1	0.60	0.305	0.59
$Error_b$	76.88	39	1.97	—	—
Within Groups	123.50	41	—	—	—
Measures	10.25	1	10.25	3.55	0.063
Groups X Measures	0.725	1	0.725	0.25	0.6247
$Error_w$	112.52	39	2.885		
Total	200.98	81	—	—	—

TABLE 17

SPLIT-PLOT REPEATED MEASURES TEST OF HYPOTHESIS 4b

Source	SS	df	MS	F	p
Between Subjects	133.78	40	—	—	—
Groups	0.061	1	0.061	0.018	0.889
$Error_b$	133.72	39	3.428	—	—
Within Groups	91.50	41	—	—	—
Measures	3.52	1	3.52	1.69	0.198
Groups X Measures	6.69	1	6.69	3.21	0.0775
$Error_w$	81.29	39	2.08	—	—
Total	225.28	81	—	—	—

F. Summary

The inferential statistical results for the hypotheses generated for the principles of conditional logic are summarized in Figure 10. Note that only Hypothesis 3b (Inversion under the biconditional interpretation) can be rejected. Here, there is a significant effect favoring the experimental group's performance on the CCRT Inversion items.

The table also shows that for the Contraposition principle, both groups did significantly better on the SSCRT than they did on the CCRT items. Also, for the Inversion principle, under the biconditional interpretation, the experimental group did significantly better on the CCRT items than they did on the SSCRT items and the control group did significantly better on the SSCRT items than they did on the CCRT items.[27]

				Statistically Significant Results (at the 0.05 level)			
			Reject Hypothesis	Groups	Measures	Interaction (Groups X Measures
VALIDITY PRINCIPLES	Forward Conditional	MC/BC	Hypothesis 1	no	n.s.	n.a.	n.a.
	Contra-position	MC/BC	Hypothesis 2	no	n.s.	sig. (for SSCRT)	n.s.
FALLACY PRINCIPLES	Inversion	MC	Hypothesis 3a	no	n.s.	n.s.	n.s.
		BC	Hypothesis 3b	yes	sig. (for EG on CCRT)	sig. (for EG on CCRT and for CG on SSCRT)	sig.
	Conversion	MC	Hypothesis 4a	no	n.s.	n.s.	n.s.
		BC	Hypothesis 4b	no	n.s.	n.s.	n.s.

Figure 10. Summary Table for Principles of Conditional Logic Hypotheses ("n.s."= non-significant result; "sig."=significant result; "n.a." = not applicable; "MC" = material conditional interpretation of answers; "BC" = biconditional interpretation of answers; "MC/BC" = both interpretations give the same results)

IV

CONCLUSIONS AND DISCUSSION

A. Conclusions

Since there was no attempt to ascertain the subjects' interpretation of the logic conditional statement or the LOGO conditional branch statement, the question specifically addressed by this study is: how does the LOGO experience affect experimental group performance on tests measuring understanding of the four conditional logic principles, when the measures of these principles are scored in a material conditional and a biconditional manner?

In addition, the study sought to determine whether and how a specific part of the LOGO computer language, the conditional branch statement, affects logical performance. Also, the study examined standardized achievement test scores to see if they were affected by participation in the LOGO experience.

Clearly, the strongest kind of evidence is that provided by inferential statistics. The inferential results in this study show no statistically significant difference between the experimental and control groups on any of the logical principles when the test items were scored in the traditional mannner (i.e., material conditional). However, when the test items were rescored under a biconditional interpretation it was found that the experimental group did significantly better than the control group on the Inversion fallacy principle.

Thus, assuming that subjects interpret logical conditional statements in a biconditional manner, the LOGO experience significantly improved the experimental group's performance on the Inversion principle of conditional logic.

Following Ennis and Paulus (1965), two levels of mastery of logical principles were examined. To meet the <u>sufficient condition</u> of mastery, a subject must score 5 or 6 correct answers out of a maximum of 6 for a principle. To meet the <u>necessary condition</u> of mastery, a subject must score at least 4 correct answers out of a maximum of 6 for a principle. Tables 6, 7, 8 and 9 contain data on mastery. For all but the Forward Conditional and the Conversion principle (under the biconditional interpretation), the percentage of the experimental group mastering a principle was equal to or greater than the percentage of control group mastery. For the Inversion principle, the percentage of control and experimental group subjects achieving the sufficient condition of mastery were equal but the experimental group had a higher percentage achieving necessary condition mastery.

The correlational results in Table 10 show a statistically significant positive non-zero correlation between the Inversion principle (measured by the CCRT and scored under the biconditional interpretation) and the "denying the antecedent" aspect of the LOGO Conditional-Branch Test. This aspect of the LOGO conditional branch statement can be viewed, because of its semantics, as LOGO-Inversion. Thus, performance on that aspect of the LOGO Conditional Branch Test that most closely mirrors logical Inversion has a significant positive correlation with performance on the Inversion principle itself. There are other significantly non-zero correlations.

B. Discussion

The inferential statistics suggest that the LOGO experience is Inversion specific, when it is assumed that subjects interpret the logical conditional statement in a biconditional manner. In addition, the correlational results indicate that learning the LOGO conditional branch statement correlates positively and significantly with the Inversion principle under the biconditional interpretation.

It thus appears that some learning did occur on a principle of conditional logic as a result of the LOGO experience and perhaps because of learning an aspect of the LOGO conditional branch statement. The results also suggest, that in the absence of any other significant inferential results, LOGO is Inversion specific, i.e., effects only Inversion significantly under a biconditional interpretation of the conditional statement.

Thus, if "correct" instruction in the principles of conditional logic means learning the material conditional interpretation, then the results of this study suggest that the LOGO experience provides "incorrect" indirect instruction in one principle of conditional logic: Inversion.

The goal of this study was quite limited. It set out to see whether the LOGO experience had any significant effect upon the learning of conditional principles. There was no attempt to find out just how subjects interpreted the conditional statement. If this could be determined we could then utilize a 2X2X2 analysis of variance or split-plot repeated measures experimental design: two groups (control and experimental) _by_ two subgroups (subjects who interpret the conditional statement in a material conditional manner and subjects who utilize a biconditional interpretation) _by_ two measures of each logical principle.

In this way we could answer such questions as whether there exists a significant difference between material conditional (or biconditional) interpreters who have had the LOGO experience and material conditional (or biconditional) interpreters who have not had the experience.

I focused upon the LOGO conditional branch statement because it seemed to be syntactically and semantically closest to the conditional statement.

Perhaps other aspects of the LOGO language might correlate significantly with conditional statements. Two possible candidates are mastery of computational iteration and recursion (see Statz, 1973). While some iterative and recursive procedures require LOGO conditional branch statements, others have no such requirement. Computational recursion and iteration seem to require a certain kind of logical reasoning to master successfully.

There was no direct test of which, if any, of the LOGO conditional branch statement interpretations the experimental subjects used. And since it was not known how subjects interpreted the conditional statement, we could not ask and answer these questions:

1) does a particular logical conditional statement interpretation, in and of itself, affect the subsequent interpretation of the LOGO conditional branch statement for a subject who has learned LOGO?

2) does a particular LOGO conditional branch statement interpretation, in and of itself, affect logical conditional reasoning performance?

These questions could be addressed in a 2X2X2X2 split-plot and analysis of variance experimental design: two groups (control and experimental group) by two subgroups (material and biconditional interpretations of the conditional statement) by two sub-sub-groups (implicit and explicit ELSE interpretation of the LOGO conditional branch statement) by two measures of each principle of conditional logic.

Because of the limited design of the experiment we can only speculate that if subjects interpret the conditional statement in a certain manner (and we score the questions appropriately) then the LOGO experience has such and such effects. In addition, we can only conjecture that if subjects interpret the conditional statement in a certain manner, and if subjects interpret the LOGO conditional branch statement in a certain way, then such-and-such results might appear.

Finally, a true test of whether the LOGO conditional branch statement affects the principles of conditional logic is an experiment where two groups learn LOGO. One group learns the LOGO conditional branch statement, the other does not. This might be a difficult and unrealistic experiment to carry out since the conditional branch statement is such a vital part of the LOGO programming language.

It is not clear just how widely one can generalize from the conclusions of this study. Randomization seems to have succeeded in creating non-biased sample groups within the fifth grade in the school. The two measures used to test for conditional logic principles had reasonabl. content and construct validity (Ennis and Paulus, 1965; Ennis, 1969). The LOGO Conditional Branch Test merely mimicked a terminal session with the LOGO system and can thus be said to have perfect content validity with no need for an assessment of construct validity.

Whether or not the school involved in the study is representative of any national average is difficult to say and is beyond the scope of this study to determine. Thus, it is safe to say that the conclusions of this study can be generalized to the extent to which the school and the fifth graders are representative nationally. In addition, the LOGO computer programming language is sufficiently different, in a variety of ways, from other commonly used computer languages, that generalization beyond LOGO to other IF-THEN-ELSE and standard conditional statements might not be warranted.

There are implications of this study which depend to some extent upon the extent to which the results can be generalized. For instance, it is sometimes said that learning computer programming influences (usually improves) ones logical reasoning abilities. This is a frequent argument heard for the incorporation of computer programming into school curricula and into the home as an educational tool.

Clearly this study does not support this argument to any great extent. It does, however, demonstrate that under certain specific conditions learning LOGO programming does have a statistically significant effect upon a logical ability.

In addition, this study uncovered evidence that the control group, which did not learn LOGO programming, improved significantly on reading achievement scores. The experimental group showed no such improvement. This was the only difference observed between the two groups in achievement measures. There was no way to determine whether this phenomenon was due to the LOGO experience, per se. Perhaps any group removed from the regular classroom for 30 hours over a 15 week period would have performed similarly. Such a result, however, does suggest a broader question: does learning a computer programming language have unintended side effects in the achievement, cognitive and affective domains? If so, what are they?

I hope that this study has paved the way for future improved studies that will address these and other questions associated with learning computer programming. This study has indicated that the interpretation of logical statements is an important consideration when evaluating "correct" and "incorrect" performance on measuring instruments and that interpretation of certain computer programming statements may be important when considering the effects of learning computer languages on the understanding and utilization of logical principles.

NOTES

1. The latest in a long line of professional society reports on this matter is the National Science Foundation sponsored study done by the National Council of Teachers of Mathematics (Osborne, 1980). Professional and lay populations surveyed agreed that problem solving should serve as a focus for the mathematics curricula of the 1980's. Over 95% identified problem solving as the development of logical reasoning and thinking.

2. Ennis (1962, 1969, 1975, 1976), Smith (1957) and Kneller (1966), for instance, present strong arguments for this connection. The Osborne (1980) study lends additional empirical support to this point of view. There has been precious little research addressing the effects of learning computer programming on problem solving. See Clement, Lochhead and Soloway (1979, 1980) and Soloway, Lochhead and Clement (1980).

3. I do not mean ability to understand symbolic logic as logicians do. Quite the contrary. I mean the ability to understand logical principles as reflected in everyday ordinary-language usage.

4. See the National Science Foundation sponsored technology assessment of personal computers (Nilles, 1980).

5. This phenomenon has been described by Green, with Ericson and Seidman (1980) within a theory of the logic and behavior of national educational systems. For example, Carnegie-Mellon University is making plans for all of its students to have their own personal computers by the year 1986. The theory of ~~the~~ "downward drift of the curriculum" predicts that it won't be very much longer before high school students will have their own computers too.

6. A number of studies report the limitations and failures of direct instruction in logical principles: (Hill, 1960; Ennis and Paulus, 1965; Ennis, 1969; Bergonsky and Ondrako, 1974; McAloon, 1969). A research review by Seidman (1980a, 1980b) summarizes these and other studies on logical reasoning ability. Many of these studies suggest that the teaching methods/media were inadaqnate.

7. For a highly abstract discussion of explanatory statements and causal relationships, see Chapter III ("Causality and Causal Explanation") in von Wright (1971). A broad collection of writings on conditionals and causation can be found in Sosa (1975). Scientific thinking and problem solving is hypothetico-deductive in nature (Piaget, 1957) and has as its central core, a logical framework. For instance see Inhelder and Piaget (1958). Hunt (1961) suggests that certain conditions might constitute a "match" between learning situations and learner so as to promote a more rapid transition between Piagetian stages than would ordinarily occur. Consider the transition from the concrete to formal operations stage of cognitive development. Does computer programming constitute such a "match" since it can be viewed as the concrete manipulation of abstract propositions? This notion is discussed in Seidman (1980a).

8. See Nilles (1980) for a technology assessment which includes such a forecast.

9. One need only examine TRS-80, Atari, Apple or PET microcomputer advertisements to detect this type of appeal. See, for example, T.H.E. Journal; 8, 2, February 1981.

10. Examples of the types of computer literacy arguments prevalent today can be found in Molnar (1977, 1980), Luehrmann (1979, 1980), Braun (1980a, 1980b), Aiken and Braun (1980). Also, see Anderson and Klassen (1980a, 1980b) for a bibiliography of computer literacy papers.

11. The study reported in this paper is part of a larger study by Seidman (1980a).

12. It is sometimes argued amongst logicians that the if-then conditional is not a logical truth-functional connective (under the material conditional interpretation) on the grounds that the antecedent and consequent of the conditional statement are not required to be related, that any conditional statement is true with a false antecedent and that the interpretation takes no cognizance of the subjunctive mood (i.e., contrafactual conditionals). DeLong (1970, 99) makes the point that despite the above objections, the "ultimate justification" for the material conditional truth-functional interpretation of the conditional statement "... is pragmatic; it has proven very effective in logical analysis." In addition, Quine (1972) points out that the ordinary conditional statement is really nothing more than a conditional assertion (rather than the assertion of a conditional) and that should the antecedent turn out to be false, it is as if the assertion was never made.

13. The names for these principles are from Ennis and Paulus (1965) and are not the same names that logicians use. For example, the Forward Conditional principle is called "modus ponens."

14. I use the word "standard" to distinguish the conditional branch statement illustrated in this Section from a variation introduced in another Section. BASIC, the most popular and prevalent higher-level microcomputer language contains this type of statement. (E.g., IF A=C THEN GOTO 20).

15. For instance, see ALGOL, LOGO, PASCAL and even newer versions of FORTRAN. An example from LOGO: IF :A=:C THEN GOTO 20 ELSE :X←Y + 1.

16. See Seidman (1980a).

17. Throughout this analysis, I have refered to two possible interpretations of the logical if-then conditional statement: material conditional and the biconditional. These are logical interpretations. However, there is some evidence that the conditional statement is interpreted in a non-logical manner. In a review of over 40 research studies spanning the last twenty years, Seidman (1980a) shows that a developmental thread of biconditional-like transductive reasoning runs through much of the literature on logical reasoning. However, transductive reasoning performance (which is non-logical reasoning) is identical to biconditional reasoning (logical reasoning). Seidman posits that this may be why transductive reasoning is so hard to detect. A good example of transductive reasoning research can be found in Knifong (1974). Also, see Piaget (1926; 1928) on transductive reasoning.

18. See McCarthy (1960) for an exposition of LISP.

19. Seymour Papert, one of the inventors of LOGO, is a leading proponent of this view and of teaching LOGO computer programming in grade school. See Papert (1980).

20. For instance: how are we to know whether "incorrect" answers under the material conditional scoring scheme that turn out to be "correct" answers under the biconditional scoring scheme truly represent a biconditional interpretation of the logical conditional statement? After all, this result could very well be due to a non-logical phenomenon, such as the "matching bias" posited by Evans (1972) or transductive reasoning posited by Seidman (1980a). Similar problems are encountered in trying to determine the interpretation of the conditional branch statement. Seidman (1980a) has tried to do this in an indirect manner.

21. Also, see Kerlinger (1973).

22. A review of this literature can be found in Seidman (1980a).

23. The LOGO Turtle is an electromechanical robot that is directed by the LOGO program to move about the floor. It can move forward, backwards and turn about its mid-point upon command. A pen in its belly can be lowered and raised thus giving the Turtle the ability to leave "Turtle traces." A graphical version of the Turtle is also available but was not used in this study. A new geometry has been developed around the "land" and graphical Turtle. See Abelson and diSessa (1980).

24. Numerous instruments have been developed since 1960, including: Hill, 1960; O'Brien and Shapiro, 1968; Ennis and Paulus, 1965; Ennis, 1969; Peel, 1967; Taplin, 1974; Paris, 1971; Roberge and Paulus, 1971; Howell, 1967; Gardiner, 1966; Paulus, 1967; Martens, 1967; Miller, 1968; McAloon, 1969; Carroll, 1971; Ryoti, 1973; Flener, 1974; Kodrof and Roberge, 1975; Antonok and Roberge, 1978. Most post-1965 measures take Ennis and Paulus (1965) as their model.

25. The two scoring methods are independent of one another since an "I don't know, not enough information given" response was one possible answer out of three.

26. Peel (1967) found similar results on a very different kind of reasoning test.

27. The content dimensions of these test items were analysed. No significant differences were found between the control and experimental groups with respect to content of logical statements. See Seidman (1980a).

BIBLIOGRAPHY

Abelson, H. and diSessa, A. Turtle Geometry: Computation as a Medium for Exploring Mathematics. Cambridge: M.I.T. Press, 1981.

Aiken, R. M. and Braun, Ludwig. "Into the 80's with Microcomputer-Based Learning." Computer, 13, 7, July 1980.

Anderson, R. E. and Klassen, D. L. "A Conceptual Framework for Developing Computer Literacy Instruction." Paper presented at Association for Computing Machinery Annual Meeting, Nashville, Tennessee, November 1980a.

_______. "The Computer Literacy Concept - A Bibliography." Minnesota Educational Computing Consortium, November 1980b.

Antonak, R. F. and Roberge, J. "Characterisitics of the Conditional Reasoning of Educable Retardates." American Educational Research Journal, 15, 4, Fall 1978, pp. 519-28.

Berzonsky, M. D. and Ondrako, M. "Cognitive Style and Logical Deductive Reasoning." Journal of Child Psychology, 43, 1, Fall 1974.

Braun, Ludwig. "Computers in Learning Environments: An Imperative for the 1980's." BYTE, 5, 7, July 1980a.

_______. Preface to Socio-Technological Instructional Module - Vol. III. Lexington: Ginn Custom Publishing, 1980b.

Campbell, D. T. and Stanley, J. C. Experimental and Quasi-experimental Designs for Research. Chicago: Rand McNally and Co., 1963.

Carroll, C. A. "Low Achiever's Understanding of Four Logical Inference Forms: An Analysis of Difficulties and of the Effect of Instruction." Unpublished doctoral dissertation, Columbia University, 1970.

Clement, J., Lochhead, J. and Soloway, E. "Positive Effects of Computer Programming on Students' Understanding of Variable and Equations." Proceedings of the National ACM Conference, Nashville, 1980.

_______. "Translating Between Symbol Systems: Isolating Common Difficulty in Solving Algebra Word Problems." COINS Technical Report 79-19, Department of Computer and Information Science, University of Massachusetts, Amherst, 1979.

DeLong, H. A Profile of Mathematical Logic. Reading: Addison-Wesley, 1970.

Ennis, R. H. "A Concept of Critical Thinking: A Proposed Basis for Research in the Teaching and Evaluation of Critical Thinking Ability." Harvard Education Review, 32, 1962, pp. 81-111.

_______. "Children's Ability to Handle Piaget's Propositional Logic: A Conceptual Critique." Review of Educational Research, 45, 1, Winter 1975, pp. 1-41.

_______. "An Alternative to Piaget's Conceptualization of Logical Competence." Child Development, 47, 1976, pp. 903-919.

Ennis, R. H., et al. "Conditional Logic and Children." Critical Thinking Project, Cornell University, 1969.

Ennis, R. H. and Paulus, D. "Critical Thinking Readiness in Grades 1-12 (Phase I: Deductive Reasoning in Adolescence)." Critical Thinking Project, Cornell University, 1965.

Evans, J. St. B. "On the Problem of Interpreting Reasoning Data: Logical and Psychological Approaches." Cognition, 1, 4, 1972, pp. 373-384.

Flener, F. O. "A Comparison of Reasoning with General and Singular Propositions by 5th, 7th and 9th Grade Students." Unpublished doctoral dissertation, University of Illinois, 1973.

Gardiner, W. L. "An Investigation of Understanding of the Meaning of the Logical Operators in Propositional Reasoning." Unpublished doctoral dissertation, Cornell University, 1965.

Green, Thomas F., with Ericson, D. P. and Seidman, R. H. Predicting the Behavior of the Educational System. Syracuse: Syracuse University Press, 1980.

Hill, S. "A Study of Logical Abilities in Children." Unpublished doctoral dissertation, Stanford University, 1961.

Howell, E. N. "Recognition of Selected Inference Patterns by Secondary School Mathematics Students." Unpublished doctoral dissertation, University of Wisconsin, 1965.

Hunt, J. McV. Intelligence and Experience. New York: Ronald, 1961.

Inhelder, B. and Piaget, J. The Growth of Logical Thinking from Childhood to Adolescence. New York: Basic Books, 1958.

Kerlinger, F. N. Foundations of Behavioral Research, 2nd edition. New York: Holt, Rinehart and Winston, 1973.

Kirk, R. E. Experimental Design: Procedures for the Behavioral Sciences. California: Brooks/Cole, 1968.

Kneller, G. F. Logic and Language of Education. New York: Wiley, 1966.

Knifong, J. D. "Logical Abilities of Young Children--Two Styles of Approach." Child Development, 45, 1, March 1974, pp. 78-83.

Kodroff, J. K. and Roberge, J. J. "Developmental Analysis of the Conditional Reasoning Abilities of Primary Grade Children." Developmental Psychology, 11, 1, 1975, pp. 21-8.

Luehrmann, Arthur. "Computer Literacy - What Should It Be?" Unpublished paper, 1981.

________. "Technology in Science Education." Technology in Science Education-The Next Ten Years: Perspectives and Recommendations, National Science Foundation, Directorate for Science Education, Washington, D.C., 1979.

McAloon, M. D. "An Exploratory Study in Teaching Units in Logic to Grades Three and Six." Unpublished doctoral dissertation, University of Michigan, 1969.

McCarthy, J., et al. LISP 1.5 Programmer's Manual. Cambridge: M.I.T. Press, 1960.

Martens, M. A. "Use of Selected Rules of Logical Inference and of Logical Fallacies by Senior High Students." Unpublished doctoral dissertation, University of Wisconsin, 1967.

Miller, W. "The Acceptance and Recognitiokn of Six Logical Inference Patterns by Secondary Students." Unpublished doctoral dissertation, University of Wisconsin, 1968.

Molnar, Andrew. "National Policy Toward Technological Innovation and Academic Computing." T.H.E. Journal, 4, 6, Sept/Oct. 1977.

________. "Understanding How to Use Machines to Work Smarter in an Information Society." T.H.E. Journal, 7, 1, January 1980.

Morrison, D. F. Multivariate Statistical Methods. New York: McGraw-Hill, 1967.

Nilles, Jack, et al. A Technological Assessment of Personal Computers, Vols. I, II, III. Personal Computer Assessment Project, University of Southern California at Los Angeles, 1980.

O'Brien T. C. and Shapiro, B. J. "The Development of Logical Thinking in Children." American Educational Research Journal, 5, 1968, pp. 531-542.

Osborne, Alan R. "Priorities in School Mathematics." Council of Teachers of Mathematics, 1980. ERIC ED 184891, ED 184892.

Papert, Seymour. "Teaching Children Thinking." Mathematics Teaching, 58, Spring 1972a.

________. "A Computer Laboratory for Elementary Schools." Computers and Automation, 21, 6, June 1972b.

________. Mindstorms: Children, Computers, and Powerful Ideas. New York: Basic Books, 1980.

Paris, S. G. "Comprehension of Language Connectives and Propositional Logical Relationships." Journal of Experimental Child Psychology, 16, 1973.

Paulus, D. H. "A Study of Children's Abilities to Deduce and Judge Deductions." Unpublished doctoral dissertation, Cornell University, 1967.

Peel, E. A. "A Method for Investigating Children's Understanding of Certain Logical Connectives Used In Binary Propositional Thinking." British Journal of Mathematical and Statistical Psychology, 20, Part I, May 1967, pp. 81-97.

Piaget, J. The Language and Thought of the Child. New York: Harcourt, Brace 1926.

_______. Judgment and Reasoning in the Child. New York: Harcourt, Brace, 1928.

_______. Logic and Psychology. New York: Basic Books, 1957.

Quine, W. V. Methods of Logic, 3rd edition. New York: Holt, Rinehart and Winston, 1972.

Roberge, J. J. and Paulus, D. H. "Developmental Patterns for Children's Class and Conditional Reasoning Abilities." Developmental Psychology, 4, 1971, pp. 191-200.

Roscoe, J. T. Fundamental Research Statistics. New York: Holt, Rinehart and Winston, 1969.

Ryoti, D. E. "Student Responses to Equivalent Inference Schemes in Class and Conditional Logic." Unpublished doctoral dissertation, University of Illinois, 1972.

Seidman, Robert H. "The Effects of Learning the LOGO Computer Programming Language on Conditional Reasoning in School Children." Unpublished doctoral dissertation, Syracuse University, 1980a.

_______. "Conditional Reasoning in Children: Empirical Evidence." Chapter in The Institutions and Processes of Education: Data Book. Syracuse: Syracuse University Graduate School of Education, 1980b.

Smith, B. O. "Logic, Thinking and Teaching." Educational Theory, 7, 4, October 1957.

Soloway, E., Lochhead, J. and Clement, J. "Does Computer Programming Enhance Problem Solving Activity? Some Positive Evidence on Algebra Word Problems." Unpublished paper, 1981.

Sosa, E., ed. Causation and Conditionals. London: Oxford University Press, 1975.

Statz, J. A. "The Development of Computer Programming Concepts and Problem Solving Abilities Among Ten-Year-Olds Learning LOGO." Unpublished doctoral dissertation, Syracuse University, 1973.

Taplin, J. E. "Reasoning with Conditional Sentences." Journal of Verbal Learning and Verbal Behavior, 10, 1971, pp. 219-25.

von Write, G. H. Expanation and Understanding. Ithaca: Cornell University Press, 1971.